dunetpc - Feature #14278

Add ROP support for 35t geometry

10/26/2016 07:15 AM - David Adams

Status:	Closed	**Start date:**	10/26/2016
Priority:	Normal	**Due date:**	
Assignee:	David Adams	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The latest version of larsoft (v06_12_00) adds support for ROPs (readout planes) to its the geometry interface. These call methods in ChannelMap35OptAlg that Gianluca has filled in with temporary exception-raising code. I have been trying to update that code but am running into some surprises.

I see Tyler's email in the code. If he or anyone else can help, please let me know.

History

#1 - 10/26/2016 07:18 AM - David Adams

The surprise I reference above is that fPlanesPerAPA is set to 3 rather than 4 and so the arrays fFirstChannelInThisPlane and fFirstChannelInNextPlane are not filled as I expect.

#2 - 10/26/2016 01:20 PM - Tyler Alion

David Adams wrote:

> The surprise I reference above is that fPlanesPerAPA is set to 3 rather than 4 and so the arrays fFirstChannelInThisPlane and fFirstChannelInNextPlane are not filled as I expect.

Hi David, good point. The variables should probably be renamed to something like fViewsPerAPA and fFirstChannelInThisView. It was an extension of something like PlanesPerTPC a long time ago, and lazy of me to not be more explicit. You can see here that fFirstChannelInThisPlane and fFirstChannelInNextPlane are filled in terms of view, with 2*(the number of wires being read out on one side):
https://cdcvs.fnal.gov/redmine/projects/dunetpc/repository/revisions/develop/entry/dune/Geometry/ChannelMap35OptAlg.cxx#L133

The difference (fFirstChannelInNextPlane[2] - fFirstChannelInThisPlane[2]) encompasses both collection planes. Does this help? Happy to discuss it more.

#3 - 10/27/2016 06:46 AM - David Adams

Tyler:

Thanks for the information. I did finally puzzle this out. The variable fPlanesPerAPA is used both for the # views per APA and for the number of planes per TPC. I renamed it to fPlanesPerTPC. I also modified variables like fFirstChannelInThisPlane to index over planes instead view because this is needed for the new ROP methods.

I put these changes in a new class Dune35tChannelMapAlg that will eventually replace ChannelMap35Alg. I also modified DUNEGeometryHelper to add an option argument that allows explicit specification of the mapping class so we can swap the old and new maps in fcl.

I am putting these changes in dunetpc now.

#4 - 10/28/2016 12:15 PM - David Adams

I have finished Dune35tChannelMapAlg. All the new ROP methods are filled in and I have added a test test_Geometry_Dune35t that exercises the mappings from that class and verified it gives the same results as ChannelMap35OptAlg which is the current default for production. The changes are in the develop branch of dunetpc.

If there are no objections, next week, I will modify DUNEGeometryHelper so that Dune35tChannelMapAlg replaces ChannelMap35OptAlg as the default for production.

#5 - 10/28/2016 05:04 PM - David Adams

I am afraid the wire mapping is not correct with the new mapping. Both the new tests and Gianluca's update of his tests show problems with channel 400. Presumably because of the array changes described above. I will try to resolve this by Monday at the latest.

#6 - 10/31/2016 11:53 AM - David Adams

The channel mapping problem is fixed. I verify that the new mapping (Dune35tCannelMapAlg) now gives the same channel-wire mapping as the old code (ChannelMap35tOldAlg).

Gianluca, could you verify your tests also pass now?

#7 - 11/01/2016 10:44 AM - Gianluca Petrillo

My test is failing at a different point now. I was in the process of changing geometry details, and now I am checking signal types more carefully.
I receive the warning Channel 400 (400) not given signal type., emitted from ChannelMapAlg::SignalType implementation.
My test then fails with the message: Geometry service claims channel #400 to be of type 0 but that the plane of C:0 T:1 P:2 W:0 is of type 1.
Specifically, the test starts from a wire ID (C:0 T:1 P:2 W:0):

- the first figure comes from this path: wireID -> (base class) -> plane ID -> ChannelMapAlg::PlaneWireToChannel -> channel (400) -> ChannelMapAlg::SignalType -> signal type (0)
- the second figure comes from this path: wire ID -> WireID::planeID -> plane ID -> GeometryCore::WirePlaneToROP -> ROPID -> ChannelMapAlg::FirstChannelInROP -> channel ID -> ChannelMapAlg::SignalType -> signal type (1)

Given that plane 2 is a collection plane, the right answer should be actually signal type 1 (geo::kCollection).
From a dump of channels vs. wires (see test dump_channel_map_dune35t_test, using the old mapping), it seems that the wire-to-channel mapping (PlaneWireToChannel) gives the correct answer. A problem might lie in ChannelMapAlg::FirstChannelInROP or GeometryCore::WirePlaneToROP.
I see also another problem: in case of failure, DUNE ChannelMapAlg::SignalType implementations return geo::kInduction, while they should return geo::kMysteryType.
Full output:

```
%MSG-i GeometryTest:  main 01-Nov-2016 09:53:27 CDT Initialization GeometryTestAlg.cxx:294
test channel to plane wire and back ...
%MSG
%MSG-w BadChannelSignalType:  main 01-Nov-2016 09:53:27 CDT Initialization
Channel 400 (400) not given signal type.
%MSG
%MSG-w GeometryTest:  main 01-Nov-2016 09:53:27 CDT Initialization
exception caught:
---- BadChannelLookup BEGIN
  Geometry service claims channel #400 to be of type 0 but that the plane of C:0 T:1 P:2 W:0 is of type 1
---- BadChannelLookup END
```

Once more: note that I am changing the test, so this output might be different from the one of the old test. Nevertheless, plane 2 being collection is, as far as I know, a fact.

I am probably explaining this in a very confusing way... please ask if you need more (or better) information.

#8 - 11/01/2016 12:14 PM - David Adams

Thanks for the information. I expanded my test coverage and saw the same problem. It is now fixed. Please try again with the develop branch of dunetpc.

The code is simplified for these and other methods.

#9 - 11/01/2016 01:11 PM - David Adams

I have added tests of NearestWireID and WireCoordinate to test_Geometry_Dune35t and have verified that the new code produces the same results as the old.

I **think** the test now covers all it should but look forward to hearing from Gianluca about anything I may have missed.

#10 - 11/02/2016 12:23 PM - Gianluca Petrillo

Here is a new one:

```
%MSG-i GeometryCore:  main 02-Nov-2016 12:18:57 CDT Initialization
Debugging information:
  - wire ID: C:0 T:2 P:0 W:0
  - channel: 512
  - SignalType(512): 0
  - WirePlaneToROP({ C:0 T:2 P:0}): C:0 S:1 R:2
  - FirstChannelInROP({ C:0 S:1 R:2 }): 800
  - SignalType(800): 1
  - SignalType({ C:0 T:2 P:0}): 1
%MSG
%MSG-w GeometryTest:  main 02-Nov-2016 12:18:57 CDT Initialization
exception caught:
---- BadChannelLookup BEGIN
  Geometry service claims channel #512 to be of type 0 but that the plane of C:0 T:2 P:0 W:0 is of type 1
```

```
---- BadChannelLookup END
```

I do not understand the logic in Dune35tChannelMapAlg::WirePlaneToROP(). It seems it returns an invalid TPC set if the TPC is above 4, and my understanding is DUNE 35t has 8 TPCs. Maybe worth a check? Definitely worth adding some explanation on the mapping in the comment (or in Dune35tChannelMapAlg class Doxygen documentation).
My understanding is that readout planes 2 and 3 are collection planes.
Also, the "readout" IDs inherit a (poor) design choice that has the validity of the ID separately stored. To return an invalid ID, the isValid member should be set to false, and setting the single TPC set number to invalid is good practice but not sufficient.

#11 - 11/02/2016 12:51 PM - David Adams

I agree that Dune35tChannelMapAlg::WirePlaneToROP was bad. I am fixing it.

I test through the geometry and apparently don't cover this method. Where is it called? If nowhere, we should drop it.

I don't see any isValid member in larcoreobj/SimpleTypesAndConstants/readout_types.h. I just updated from git.

#12 - 11/02/2016 01:07 PM - David Adams

I have pushed the fix for Dune35tChannelMapAlg::WirePlaneToROP to dunetpc develop. Please test again.

#13 - 11/02/2016 01:33 PM - Gianluca Petrillo

David Adams wrote:

> I don't see any isValid member in larcoreobj/SimpleTypesAndConstants/readout_types.h. I just updated from git.

It's inherited from readout::CryostatID, that is geo::CryostatID .

#14 - 11/02/2016 01:54 PM - Gianluca Petrillo

David Adams wrote:

> I agree that Dune35tChannelMapAlg::WirePlaneToROP was bad. I am fixing it.
>
> I test through the geometry and apparently don't cover this method. Where is it called? If nowhere, we should drop it.

Well. it being new, it's not surprising that it's not very used.
I am using it now to implement SignalType(geo::PlaneID), using plane -> ROP -> channel and then SignalType(raw::ChannelID_t).

Anyway: DUNE 35t new mapping now now **passes** the standard geometry test.
DUNE Far Detector mapping reports instead ChannelToROP not implemented.

#15 - 11/02/2016 01:58 PM - Gianluca Petrillo

Related: is the new mapping going to be the official one for DUNE 35t?
When that is official, I would like to push the version of the test using this mapping instead of the old one.

#16 - 11/02/2016 02:03 PM - David Adams

Glad to hear we finally pass.

Let me know when the new geometry method is added and I will add it to my test.

I have created a separate issue for adding ROP support for FD. Please see #14362.

I will wait a day before making Dune35tChannelMapAlg the default for 35t.

#17 - 11/10/2016 07:21 AM - David Adams

Analogous to the work done here for 35t, issue #14362 describes how DuneAPAChannelMapAlg was created as a new class for mapping FD and protoDUNE channels. I would like to consolidate mappers an also use it in place of Dune35tChannelMapAlg described here. I have verified with test_Dune35tGeometry that it gives the same TPC results for 35t but there is no checking of the photon detectors and I do see some code differences there. I have added Alex to the watcher list here so he can help answer questions. Please add anyone else who is relevant.

First, I see these methods:

```
  unsigned int NOpChannels(unsigned int NOpDets)                        const;
  unsigned int NOpHardwareChannels(unsigned int opDet)                  const;
  unsigned int OpChannel(unsigned int detNum, unsigned int channel = 0) const;
  unsigned int OpDetFromOpChannel(unsigned int opChannel)               const;
  unsigned int HardwareChannelFromOpChannel(unsigned int opChannel)     const;
```

I see the implementation is a bit different for 35t which has different channel/detector counts depending on detector index. Should I keep this mapping? If so, should I use the detector name to decide which mapping to use?

Thanks for any guidance.

#18 - 11/10/2016 08:54 AM - David Adams

I have extended the geometry tests test_GeometryDune35t, test_GeometryDune10kt and test_GeometryProtoDune to cover the optical methods listed in the preceding note for this issue.

#19 - 11/10/2016 10:02 AM - Alexander Himmel

The 35ton mapping should be kept for the 35ton, and kept distinct from the FD/protoDUNE mapping which is different. I **think** detector name can work as well as channel number for doing the mapping, but I do not remember exactly what that mapping would be. You'll need to poke around to figure out what corresponds to what.

#20 - 11/10/2016 10:38 AM - David Adams

Thanks Alex. I have now modified DuneApaChannelMapAlg so that it uses the 35t OpDet mapping if the DetectorVersion begins with "dune35t" and have verified this provides the same mapping as Dune35tChannelMapAlg and ChannelMap35OptAlg.

#21 - 11/10/2016 01:45 PM - David Adams

- Status changed from Assigned to Closed

As noted in #14362, the new channel map DuneApChannelMapAlg is now used by default for 35t. I close this issue. Please report any problems at #14362.

#22 - 11/10/2016 02:50 PM - David Adams

I have removed Dune35tChannelMapAlg from dunetpc.